

数据可视化

Code ▾

This is an R Markdown (<http://rmarkdown.rstudio.com>) Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Cmd+Shift+Enter*.

Hide

```
library(tidyverse)
library(mosaic)
library(ggplot2)
library(moments)
library(sjlabelled)
```

步骤 2：查看一共有多少个国家

从RDS文件中读取数据，RDS（R DataSerialization）文件是一种二进制文件格式，常用于保存R对象和数据。通过readRDS()函数，我们可以将RDS文件中的数据加载到R的工作环境中，并将其赋值给变量WVS。然后，将WVS数据框中名为V2A的列转换为字符型。并赋值给原来的那一列。接下来找出没有重复的国家编号，存在一个矩阵中，求这个矩阵的长度。于是就得到了国家编号的个数。

Hide

```
WVS <- readRDS("WVS.rds")
WVS$V2A <- as.character(WVS$V2A)
num_countries <- length(unique(WVS$V2A))
print(paste(" 共有", num_countries, " 个国家编号"))
```

```
[1] " 共有 61 个国家编号"
```

步骤 3：写代码筛选出国家编码变量（V2A）和生活满意度变量（V23）。

Hide

```
selected_data <- WVS[, c("V2A", "V23")]
life_satisfaction <- selected_data$V23
country_code <- selected_data$V2A
```

步骤 4：分析该变量的分布特征

写代码得到生活满意度变量的 median, mean, standard deviation, skewness, kurtosis 统计量。

median: 中位数、mean: 平均值、standard deviation: 标准差, skewness: 偏度, kurtosis: 峰度。
中位数 (Median) : 将数据按升序排列后, **位于中间位置的数值**。它代表了**数据的中心位置**, 不受极值的影响。 平均值 (Mean) : 将所有数据求和后再除以数据的个数, 用于**衡量数据的集中趋势**。

标准差 (Standard Deviation) : 衡量数据分散程度的一种度量。它测量了**数据与其平均值之间的差异程度**。 偏度 (Skewness) : 用于描述数据分布的**对称性和偏斜程度**。正偏斜表示数据右侧尾部较长, 负偏斜表示数据左侧尾部较长。偏度为0意味着数据接近对称。 峰度 (Kurtosis) : 用于描述数据分布的**峰态**。正峰度表示数据比**正态分布更陡峭**, 负峰度表示数据比正态分布更平缓。峰度为0表示数据与正态分布具有相似的峰态。

Hide

```
# 加载 moments 库,以引用skewness函数计算偏度
library(moments)

# 计算统计量
median_val <- median(WVS$V23, na.rm = TRUE)
mean_val <- mean(WVS$V23, na.rm = TRUE)
sd_val <- sd(WVS$V23, na.rm = TRUE)
skewness_val <- skewness(WVS$V23, na.rm = TRUE)
kurtosis_val <- kurtosis(WVS$V23, na.rm = TRUE)
# 打印统计量
cat("中位数:", median_val, "\n")
```

中位数: 7

Hide

```
cat("平均值:", mean_val, "\n")
```

平均值: 6.779668

Hide

```
cat("标准差:", sd_val, "\n")
```

标准差: 2.365454

Hide

```
cat("偏度:", skewness_val, "\n")
```

偏度: -0.7579571

Hide

```
cat("峰度:", kurtosis_val, "\n")
```

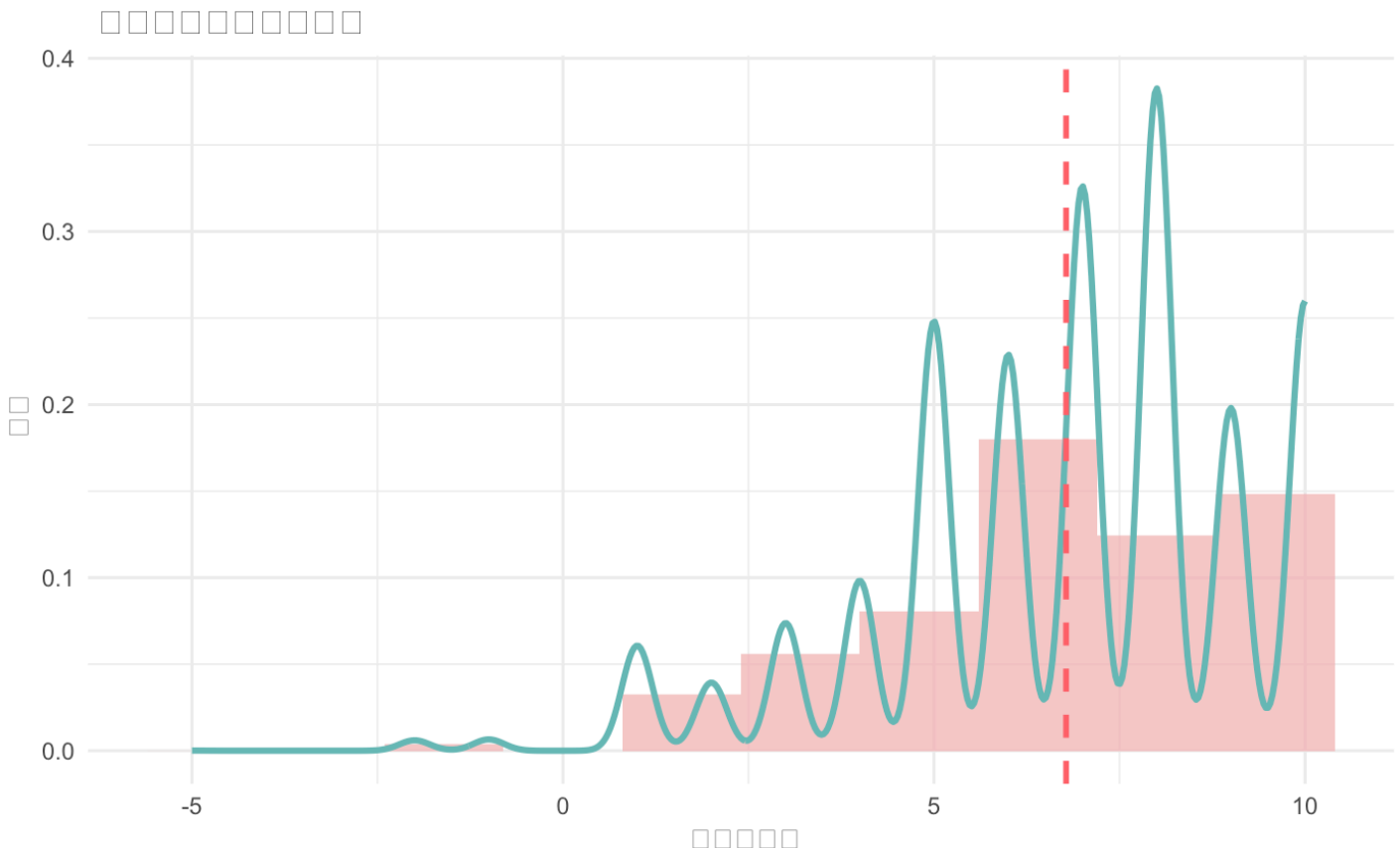
峰度: 3.422007

步骤 5: 绘制直方图

绘制直方图来说明生活满意度变量的分布，在直方图上添加密度曲线、添加一条垂线来展示均值，如果需要可以调整柱体宽度(bin width)，通过直方图对变量的分布特征进行文字解读。

[Hide](#)

```
library(ggplot2)
plot <- ggplot(WVS, aes(x = V23)) +
  geom_histogram(aes(y = after_stat(density)), binwidth = 1.6, fill = "#f4aeb0", alpha = 0.7) +
  geom_density(color = "#65bab7", linewidth = 1.1) +
  geom_vline(aes(xintercept = mean(V23, na.rm = TRUE)), color = "#fe616a", linetype="dashed", linewidth = 1) +
  labs(title = "生活满意度分布直方图", x = "生活满意度", y = "密度") +
  theme_minimal()
print(plot)
```



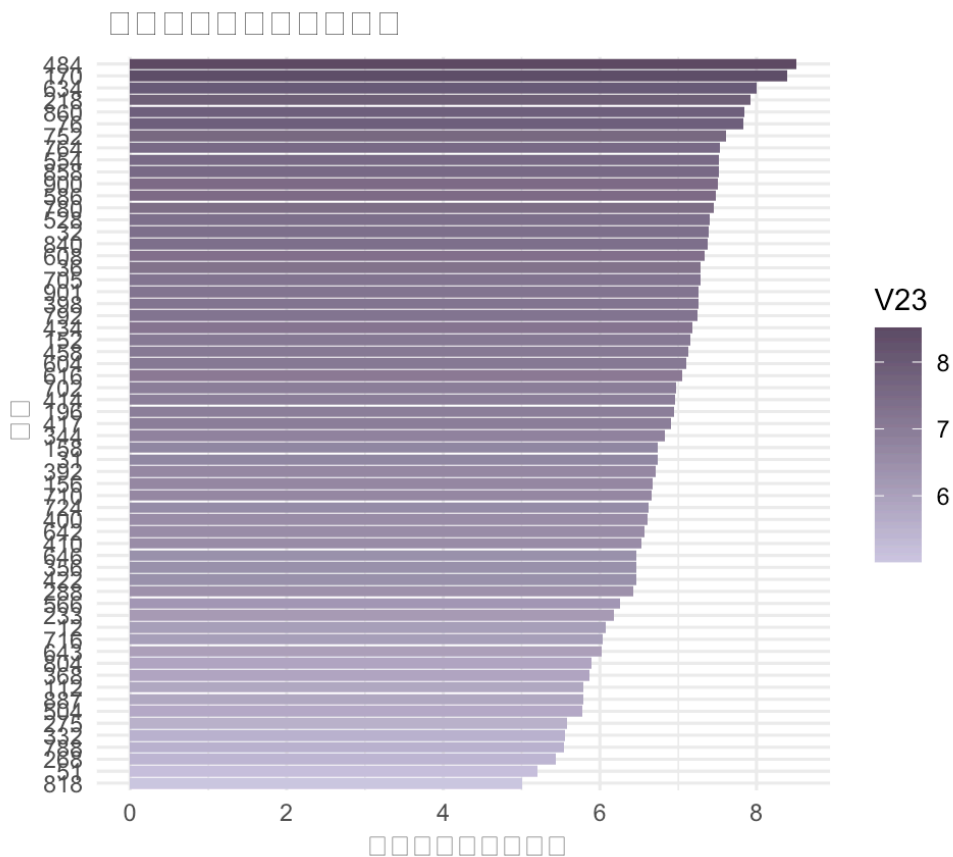
步骤 6: 统计每个国家人民的平均生活满意

度分数

统计每个国家人民的平均生活满意度分数，从低到高图示这些国家。回答问题：哪一个国家的人民生活满意度最高？哪一个最低呢？

[Hide](#)

```
# 计算每个国家的平均生活满意度
country_satisfaction <- aggregate(V23 ~ V2A, WVS, mean, na.rm = TRUE)
# 对数据按生活满意度排序
country_satisfaction <- country_satisfaction[order(country_satisfaction$V23),]
# 使用 ggplot2 绘图
plot_country <- ggplot(country_satisfaction, aes(x = reorder(V2A, V23), y = V23, fill = V23)) +
  geom_bar(stat = "identity") + coord_flip() + labs(title = " 各国生活满意度平均分数", x = " 国家", y = " 平均生活满意度分数") + theme_minimal() +
  scale_fill_gradient(low = "#cdc7e2", high = "#5e4c66") + # 从浅蓝到深蓝的渐变
  theme(aspect.ratio = 1) # 设置宽高比为 1
print(plot_country)
```


[Hide](#)

```
# 获取满意度最高和最低的国家
highest_country <- tail(country_satisfaction, 1)
lowest_country <- head(country_satisfaction, 1)
cat(" 人民生活满意度最高的国家是:", highest_country$V2A, " 分数为:", highest_country$V23,
    "\n")
```

人民生活满意度最高的国家是：484 分数为：8.5125

Hide

```
cat(" 人民生活满意度最低的国家是:", lowest_country$V2A, " 分数为:", lowest_country$V23,
    "\n")
```

人民生活满意度最低的国家是：818 分数为：5.010506

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Cmd+Option+I*. When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Cmd+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.